

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-91297

(P2002-91297A)

(43) 公開日 平成14年3月27日 (2002.3.27)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テ-マ-ト (参考)
G 0 9 C 1/00	6 1 0	G 0 9 C 1/00	6 1 0 B 5 J 1 0 4
	6 5 0		6 5 0 Z

審査請求 未請求 請求項の数11 O L (全 17 頁)

(21) 出願番号 特願2001-207326 (P2001-207326)  
(22) 出願日 平成13年7月9日 (2001.7.9)  
(31) 優先権主張番号 特願2000-212813 (P2000-212813)  
(32) 優先日 平成12年7月13日 (2000.7.13)  
(33) 優先権主張国 日本 (J P)

(71) 出願人 000005223  
富士通株式会社  
神奈川県川崎市中原区上小田中4丁目1番  
1号  
(72) 発明者 下山 武司  
神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内  
(72) 発明者 伊藤 孝一  
神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内  
(74) 代理人 100074099  
弁理士 大菅 義之 (外1名)

最終頁に続く

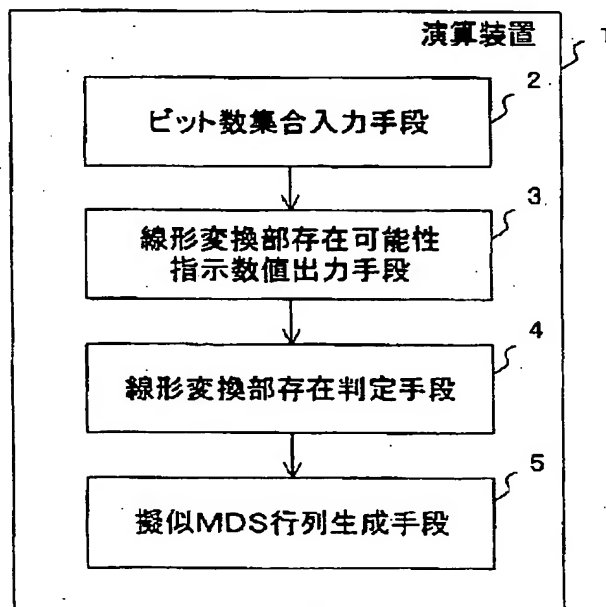
(54) 【発明の名称】 F関数内部にSPN構造を用いた演算装置および演算方法

(57) 【要約】

【課題】 F関数内部のSPN構造中の複数のSボックスの間で入出力ビット数が同一でない場合に、SPN構造内の線形変換部としてデータ拡散性能に優れたものを用いて演算を行う。

【解決手段】 演算装置1に与えられる入力データの全ビット数を非均等に分割したビット数集合Tの入力を受け取る手段1と、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値 $A_T$ を出力する手段3を備える。さらに $A_T$ の値が正のとき適切な線形変換部が存在すると判定する手段4と、そのような線形変換部としての擬似MDS行列を生成する手段5を備える。

本発明の原理構成ブロック図



## 【特許請求の範囲】

【請求項1】 複数のSボックスと線形変換部とを備えるSPN構造をF関数の内部に用いた演算装置において、

該演算装置に与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T = \{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取るビット数集合入力手段と、

該分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値 $A_T$ を出力する線形変換部存在可能性指示数値出力手段とを備えることを特徴とするF関数内部にSPN構造を用いた演算装置。

【請求項2】 前記線形変換部存在可能性指示数値出力手段が、

前記集合 $T$ の要素から任意の $k$ 個を選んで生成した集合の要素の和の最小値 $u_k$  ( $k=1, 2, \dots, r$ )を求める最小値決定手段と、

集合 $T$ の要素から任意の $k$ 個を選んで生成した集合の要素の和の最大値 $v_k$  ( $k=1, 2, \dots, r$ )を求める最大値決定手段とを更に備え、

数値 $k$ に対して $u_k \geq v_{k'}$  ( $k'=0, 1, \dots, r, v_0=0$ )を満たす $k'$ の最大値を $k$ から減算した値を $w_k$  ( $k=1, 2, \dots, r$ )とし、 $w_k$ の最大値を $(r+1)$ の値から減算して前記 $A_T$ の値を求めることを特徴とする請求項1記載のF関数内部にSPN構造を用いた演算装置。

【請求項3】 前記演算装置において、

前記 $A_T$ の値が正か正でないかを判定し、正である時前記適切な線形変換部が存在すると判定する線形変換部存在判定手段を更に備えることを特徴とする請求項1、または2記載のF関数内部にSPN構造を用いた演算装置。

【請求項4】 前記演算装置において、

前記線形変換部が存在すると判定された時、該線形変換部として、前記ビット数分割が均等に行われた場合のMDS行列に対応する擬似MDS行列を生成する擬似MDS行列生成手段を更に備えることを特徴とする請求項3記載のF関数内部にSPN構造を用いた演算装置。

【請求項5】 前記擬似MDS行列生成手段が、要素が0、または1の $t_i$ 行、 $t_j$ 列の部分行列 $M_{ij}$ を要素として、 $r$ 行、 $r$ 列に並べた行列を $M = (M_{ij})$  ( $i=1, 2, \dots, r, j=1, 2, \dots, r$ )として、 $e=1$ から $(A_T - 1)$ までの各正数に対して $c(e) = e + r - A_T + 1$ を求め、集合 $T$ の要素を $e$ 個任意に選んだ集合 $T_1 = \{t_{i1}, t_{i2}, \dots, t_{ie}\}$ と、要素を $c(e)$ 個任意に選んだ集合 $T_2 = \{t_{j1}, t_{j2}, \dots, t_{jc(e)}\}$ を求め、該集合 $(T_1, T_2)$ に対応する任意のあらゆる $M$ の小行列、および集合 $(T_2, T_1)$ に対応する任意のあらゆる $M$ の小行列

の階数の値が、それぞれ自小行列の行数、または列数のいずれかに等しい行列 $M$ を求めることを特徴とする請求項4記載のF関数内部にSPN構造を用いた演算装置。

【請求項6】 前記集合 $(T_1, T_2)$ に対応する小行列は、前記行列 $M = (M_{ij})$ を構成する前記 $r$ 行、 $r$ 列の要素としての部分行列 $M_{ij}$ のうちで、前記 $t_{i1}, t_{i2}, \dots, t_{ie}$ にそれぞれ対応する行と、 $t_{j1}, t_{j2}, \dots, t_{jc(e)}$ にそれぞれ対応する列とによって指定される部分行列によって構成されることを特徴とする請求項5記載のF関数内部にSPN構造を用いた演算装置。

【請求項7】 複数のSボックスと線形変換部とを備えるSPN構造をF関数内部に用いる演算方法において、与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T = \{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取り、

該分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値 $A_T$ を出力することを特徴とするF関数内部にSPN構造を用いた演算方法。

【請求項8】 前記演算方法において、前記 $A_T$ の値が正か正でないかを判定し、正である時前記適切な線形変換部が存在すると判定することを特徴とする請求項7記載のF関数内部にSPN構造を用いた演算方法。

【請求項9】 前記線形変換部が存在すると判定された時、該線形変換部として、前記ビット数分割が均等に行われた場合のMDS行列に対応する擬似MDS行列を生成することを特徴とする請求項8記載のF関数内部にSPN構造を用いた演算方法。

【請求項10】 複数のSボックスと線形変換部とを備えるSPN構造をF関数内部に用いた演算を実行する計算機によって使用される記憶媒体において、与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T = \{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取るステップと、

該分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値 $A_T$ を出力するステップとを計算機に実行させるためのプログラムを格納した計算機読み出し可能可搬型記憶媒体。

【請求項11】 複数のSボックスと線形変換部とを備えるSPN構造をF関数内部に用いた演算を実行する計算機によって使用されるプログラムにおいて、与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T = \{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取る手順と、該分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値 $A_T$ を出力する手順とを計算機に実行さ

せるためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は共通鍵ブロック暗号化方式に係り、さらに詳しくはFeistel 構造と呼ばれる構造内のF関数の内部で用いられる複数のSボックスに対する入出力ビット数が、複数のSボックスの間で同一でない場合に、複数のSボックスの後に備えられる線形変換部として、データ拡散性のよい線形変換部を生成し、その線形変換部を用いて入力データに対する暗号化を行う暗号化装置、および暗号化方法に関する。

【0002】

【従来の技術】高度情報化社会を迎え、情報セキュリティの確保は緊急の課題となっている。情報セキュリティの基本となるのはデータの暗号化である。高度情報化社会において高速、かつ安全な通信を実現するために、共通鍵ブロック暗号は不可欠の技術である。この共通鍵ブロック暗号のアルゴリズムとして、例えば応用分野に応じて様々な方式が提案されているが、その1つとしてFeistel 構造と呼ばれる単純な繰返し構造のアルゴリズムがある。

【0003】図15はこのFeistel 構造が16段繰り返されたDES暗号方式の説明図である。同図において入力P、例えば64ビットは右側32ビットと左側32ビットとに分割され、右側32ビットはF関数51と呼ばれる非線形関数に入力され、その出力と左側32ビットとがXOR52によって排他的論理和がとられ、その結果は右側32ビットとして次の段に与えられ、次の段への左側32ビットとしては入力64ビットのうち右側32ビットがそのまま与えられる。

【0004】図16は図15内のF関数51の構成例である。入力、例えば32ビットはビット拡張部E61によって48ビットに拡張され、XOR62によってその48ビットと鍵K<sub>i</sub> 48ビットとの排他的論理和がとられ、その出力は6ビットずつに分割されてSボックスと呼ばれる非線形関数にそれぞれ入力される。各Sボックス63の出力は、例えば4ビットとされ、合計32ビットが線形関数P64に入力され、データの拡散が行われる。このような構造は一般にSPN（サブスティテューションパーミューテーションネットワーク）構造と呼ばれる。

【0005】Sボックスは暗号装置の非線形の攪拌出力を得るため、またSボックスに続いて行われる線形関数Pは、Sボックスによる局所的な非線形出力をデータ全体に拡散させるために用いられるが、暗号装置に組み込まれる上で拡散性のよい線形変換とは何か、具体的にどう求めるか、という研究が従来より行われている。一般に、暗号に用いられる線形変換としては、一つのSボックスの出力が次のラウンドにおいて、出来る限り多くのSボックスの入力に関係することが望ましいが、現在で

はより拡張された線形関数として、次のような性質を満たすものがよいとされている。すなわち、Sボックスの入出力ビット数sに対して、線形変換Pの入力Xおよび出力Yをsビット単位、t個のブロック $X = (x_1, \dots, x_t)$ ,  $Y = (y_1, \dots, y_t)$ , (各 $x_i, y_j$ はsビット)に分割した場合、

$$Y = P(X)$$

の入出力間で成立する任意の線形関係式

$$f(x_1, \dots, x_t, y_1, \dots, y_t) = 0$$

には、入出力 $x_i, y_j$  合わせて2t個の変数のうち、t+1個以上の変数が含まれている(=係数が0ではない) というものである。

【0006】このような性質を満たす線形変換PとしてMDS変換が知られている。この変換は線形変換Pにおけるデータの拡散性を定義する1つの概念としての分岐数を最大とする線形変換である。この分岐数は暗号に対する差分攻撃や、線形攻撃に対する強度を評価するパラメータであり、その詳細については次の文献で説明されている。

【0007】文献) 共通鍵ブロック暗号の選択/設計/評価に関するドキュメント、通信・放送機構

図17はそのようなMDS変換を実現する線形関数Pの説明図である。同図においては、4つのSボックス71へのそれぞれの入力、および出力は8ビットであり、合計32ビットが入力xとして線形関数Pに与えられるものとする。線形関数Pへの入力x、および出力yを、それぞれSボックスに対応させて、8ビットずつに分割した変数 $x_i$  ( $i=1\sim4$ )、 $y_j$  ( $j=1\sim4$ ) とする。

【0008】ここで $x_i$  に入力差分 $\Delta x_i$  が与えられた時、そのiの集合を次のように書き、これを入力アクティブSボックスと名付ける。

$$\{i \mid \Delta x_i \neq 0\}$$

例えば $x_1, x_2$  に入力差分が与えられた時、この集合は{1, 2}となる。

【0009】この入力アクティブSボックスに対応して、出力差分 $\Delta y_j$  が生じる $y_j$  に対応して、次の集合を出力アクティブSボックスと名付ける。

$$\{j \mid \Delta y_j \neq 0\}$$

これら2つの集合の和集合

$$\{i \mid \Delta x_i \neq 0\} \cup \{j \mid \Delta y_j \neq 0\}$$

をアクティブSボックスと名付ける。

【0010】そしてこの集合アクティブSボックスの要素の数 $\text{act } S(P)$ の最小値は、線形変換Pによって決定される。アクティブSボックスの要素の数の最小値 $\min(\text{act } S(P))$

をアクティブSボックスの数と呼ぶことにする。このアクティブSボックスの数の最大値は、前述の線形関係式に含まれる変数の数(t+1)に一致するとされている。このアクティブSボックスの数の最大値が、例えば

5となる線形変換Pが存在するとすれば、入力 $x_i$  ( $i=1\sim 4$ )の1個が変化すると、出力 $y_j$  ( $j=1\sim 4$ )の4個が変化することになり、また出力の1個は入力5個から影響されることになる。

【0011】図18はこのようなMDS変換に相当するMDS行列の説明図である。同図においてMDS行列は、それぞれ例えば0、または1の要素からなる8行、8列の部分的な行列 $a_{ij}$  ( $i=1\sim 4, j=1\sim 4$ )から構成されている。この $a_{ij}$ の行数と列数は、図17で説明したSボックス71の入出力ビット数に対応する。

【0012】このようなMDS行列が持つべき性質を説明する。図18の行列が、MDS行列として図17で説明した線形関数Pに望まれる高い拡散性を有するためには、部分的な行列 $a_{ij}$ を要素として考えた場合の4行、4列の全体から、行数と列数が等しい任意の小行列を選択した時に、その全ての小行列が正則であることが必要とされている。

【0013】すなわち例えば1行と1列を指定した(1, 1)小行列、2行と2列を指定した(2, 2)小行列、3行と3列を指定した(3, 3)小行列、および行列全体と一致する(4, 4)小行列の全てが、逆行列を持ち、同じ配置の行列式が0でなく、ランク(階数)がフルであるという性質を持つものとされている。

【0014】

【発明が解決しようとする課題】このように共通鍵ブロック暗号化方式におけるFeistel 構造内のF関数の中で、データの拡散に重要な役割を果たす線形変換PとしてのMDS行列の設計は、従来は複数のSボックスの入出力サイズが等しいことを前提として行われていたが、複数のSボックスの間で入出力サイズが異なる場合には、適切な線形変換Pが存在するか否か、存在するとすればその変換をどのように構成すればよいかについては、従来全く知られていないという問題点があった。

【0015】本発明の課題は、上述の問題点に鑑み、複数のSボックスの間で入出力サイズが異なる場合に、データの拡散性に優れた線形変換が存在するか否かを判定し、そのような線形変換が存在する場合に、そのような線形変換に相当する擬似MDS行列を生成し、それを用いて入力データに対応する暗号文を生成する暗号文生成装置、および生成方法を提供することである。

【0016】

【課題を解決するための手段】図1は本発明の演算装置の原理構成ブロック図である。同図は、Feistel 構造のF関数の内部に、複数のSボックスと線形変換部とを備える演算装置1の原理構成ブロック図である。

【0017】図1においてビット数集合入力手段2は、演算装置1に与えられる入力データの全ビット数を非均等に分割したビット数の集合 $T=\{t_1, t_2, t_3, \dots, t_r\}$ の入力を受け取るものである。

【0018】また線形変換部存在可能性指示数値出力手

段3は、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応して、データ拡散性に優れた線形変換部の存在可能性を示す値、例えばアクティブSボックスの数の最大値 $A_T$ を出力するものである。

【0019】本発明の実施の形態においては、この $A_T$ の値が正である時、適切な線形変換部が存在すると判定する線形変換部存在判定手段4を更に備えることも、またそのような線形変換部として、ビット数分割が均等に行われた場合のMDS行列に対応する擬似MDS行列を生成する擬似MDS行列生成手段5を更に備えることもできる。

【0020】また発明の実施の形態においては、線形変換部存在可能性指示数値出力手段3が、前述のビット数集合の要素から任意の $k$ 個を選んで生成した集合の要素の和の最小値 $u_k$  ( $k=1, 2, \dots, r$ )を求める最小値決定手段と、同様に $k$ 個を選んで生成した集合の要素の和の最大値 $v_k$ を求める最大値決定手段とを更に備え、数値 $k$ に対して $u_k \geq v_{k'}$  ( $k'=0, 1, \dots, r, v_0=0$ )を満たす $k'$ の最大値を $k$ から減算した値を $w_k$ とし、 $w_k$ の最大値を $(r+1)$ の値から減算して $A_T$ の値を求めることもできる。

【0021】更に本発明の実施の形態においては、擬似MDS行列生成手段5は、要素が0、または1の $t_i$ 行、 $t_j$ 列の部分行列 $M_{ij}$ を要素として $r$ 行、 $r$ 列に並べた行列を $M=(M_{ij})$  ( $i, j=1, 2, \dots, r$ )とし、 $e-1$ から $(A_T-1)$ までの各正数に対して、 $c(e)=e+r-A_T+1$ 求め、集合 $T$ の要素を $e$ 個任意に選んだ集合 $T_1$ と、要素を $c(e)$ 個任意に選んだ集合 $T_2$ を求め、その集合 $(T_1, T_2)$ に対応する任意のあらゆる $M$ の小行列、および集合 $(T_2, T_1)$ に対応する任意のあらゆる $M$ の小行列の階数の値がそれぞれ小行列の行数、または列数のいずれかに等しい行列 $M$ を求めることもできる。

【0022】この時、例えば集合 $(T_1, T_2)$ に対応する小行列は、前述の部分行列 $M_{ij}$ のうち、集合 $T_1$ の各要素にそれぞれ対応する行と、集合 $T_2$ の各要素に対応する列によって指定される部分行列によって構成されることもできる。

【0023】本発明の演算方法として、複数のSボックスと線形変換部とを備えるSPN構造をF関数の内部に用いる演算方法において、与えられる入力データのビット数を非均等に分割したビット数の集合 $T$ の入力を受け取り、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値、例えばアクティブSボックスの数の最大値 $A_T$ を出力する方法が用いられる。

【0024】この方法においては、発明の実施形態では、 $A_T$ の値が正である時、適切な線形変換部が存在すると判定することもでき、またそのような線形変換部と

して、ビット数分割が均等に行われた場合のMDS行列に対応する擬似MDS行列を生成することもできる。

【0025】更に本発明においては、複数のSボックスと線形変換部とを備えるSPN構造を、F関数の内部に用いた演算を実行する計算機によって使用される記憶媒体として、与えられる入力データの全ビット数を非均等に分割したビット数集合Tの入力を受け取るステップと、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値、例えばアクティブSボックスの数の最大値 $A_T$ を出力するステップとを計算機に実行させるためのプログラムを格納した計算機読出し可能可搬型記憶媒体が用いられる。

【0026】また更に本発明においては、複数のSボックスと線形変換部とを備えるSPN構造を、F関数の内部に用いた演算を実行する計算機によって使用されるプログラムとして、与えられる入力データの全ビット数を非均等に分割したビット数集合Tの入力を受け取る手順と、分割されたビット数をそれぞれ入・出力ビット数とする複数のSボックスに対応する適切な線形変換部の存在可能性を示す値、例えばアクティブSボックスの数の最大値 $A_T$ を出力する手順とを計算機に実行させるためのプログラムが用いられる。

【0027】以上説明したように、本発明によればFeistel構造の内部のF関数を構成するSPN構造内で、複数のSボックスの入出力ビット数が非均等の場合に対して、データの拡散性に優れた線形変換部の生成が可能となる。

【0028】

【発明の実施の形態】本発明においては、Feistel構造の内部に備えられるF関数を構成するSPN構造内で、複数のSボックスの間で入出力ビット数が全て同じではない場合の暗号化アルゴリズム、およびそのアルゴリズムを用いた暗号化装置を本発明の実施形態として説明する。

【0029】図2はそのような暗号化装置の構成ブロック図である。同図において暗号化装置は処理装置10、入力ファイル11、出力ファイル12、表示装置13、および入出力装置14によって構成されている。

【0030】入力ファイル11には、例えば暗号化の対象としての平文、Feistel構造内のF関数への入力データのビット数 $n$ 、ビット数 $n$ が複数のSボックスに入力される場合の各Sボックスへの入力ビット数 $t_1, t_2, \dots, t_r$ を要素とする集合Tなどが格納されている。

【0031】処理装置10の内部には、入力ファイル11に格納されている集合Tの内容を用いて、複数のSボックスに対するそれぞれの入出力ビット数が同じでない場合に、その複数のSボックスの出力に対応する適切な線形変換部の存在可能性を示す数値 $A_T$ を計算する $A_T$

計算部15、計算された $A_T$ の値を用いてそのような線形変換部が存在するか否かを判定する線形変換部存在判定部16、そのような線形変換部が存在すると判定された時に、そのような変換部としての擬似MDS行列を計算する擬似MDS行列生成部17、生成された擬似MDS行列を用いて、入力ファイル11に格納されている平文に対する暗号文を生成する暗号文生成部18などを備えている。

【0032】出力ファイル12には、 $A_T$ 計算部15によって計算された $A_T$ の値、擬似MDS行列、およびその擬似MDS行列を用いた暗号化アルゴリズムなどが格納される。

【0033】図3は本実施形態において用いられるF関数の内部のSPN構造の例である。入力データ32ビットは、例えば6, 5, 5, 5, 5, および6ビットに分割され、非線形変換部としてのそれぞれのSボックス21に入力される。各Sボックスは入力ビット数と同じ出力ビット数を持ち、各Sボックスの出力は合成されて32ビットとして線形変換部P22に与えられ、その変換結果がF関数の出力となる。

【0034】本実施形態においては、このように複数のSボックスへの入出力ビット数が同じでない場合に、そのビット分割の仕方によって適切な線形変換部Pが存在するか否か、また存在する場合にはその線形変換部をどのように求めるかが発明のポイントとなる。

【0035】ここで入力データのビット数 $n$ を非均等に分割する理由について説明する。例えば従来技術で説明した図17では、入力32ビットを4つに分割した8ビットずつが、4つのSボックス71に入力されている。このようなSボックスは一般的には演算の高速化のために計算機の一次キャッシュメモリにテーブルとして格納され、そのテーブルにアクセスすることによって演算が行われる。図17ではテーブルは4つであり、4回のテーブルアクセスが必要となる。

【0036】これに対して本実施形態では、図3に示すように例えば、入力32ビットが6, 5, 5, 5, 5, および6ビットの6つに分割され、6個のSボックスにそれぞれ入力される。このようにビット数の少ない6個のSボックスに分割すると、それぞれのSボックスに対応するテーブルのサイズは小さくなり、一次キャッシュメモリ量の少ない計算機を使用しても、演算を実行することが可能となる。

【0037】最近の計算機の一次キャッシュメモリ量の増大の傾向につれて、1つのテーブルサイズを大きくしてテーブルアクセスの回数を減らし、演算を高速化する可能性が開かれている。そこで本実施形態においては、計算機の一次キャッシュメモリ量に対応して、ビット数分割を変更できるようなビット数分割法を用いることにする。

【0038】前述のように32ビットを8ビット×4に

分割している場合には、テーブルの数を3つにするためには、8、16、8ビットというような分割に変換するか方法がなく、16ビット入力のSボックスに対しては $2^{16}$ 個の領域を持つテーブルが必要となってしまうことになる。それに対して図3の分割方法では、例えば2個ずつまとめて11、10、11ビットの3つに分割することもでき、 $2^{11}$ 個の領域を持つテーブルを計算機の一次キャッシュメモリに格納することができれば、演算の高速化が可能となる。

【0039】図4は本実施形態における暗号文生成処理の全体フローチャートである。同図において処理が開始されると、まずステップS1で図2で説明した線形変換部が存在するか否かを判定するための数値 $A_T$ が求められる。この数値 $A_T$ としては、前述のアクティブSボックスの要素の数の最小値の最大値が用いられる。以後この $A_T$ を“アクティブSボックス数の最大値”と呼ぶ。

【0040】そしてステップS2で、求められた $A_T$ の値に応じて、適切な線形変換Pが存在するか否かが判定される。具体的には $A_T$ の値が正の数である時にはそのような線形変換が存在すると判定され、0、または負の数である時にはそのような線形変換は存在しないと判定される。

【0041】線形変換が存在すると判定されると、ステップS3でその線形変換を実現する行列、すなわち擬似MDS行列が生成され、ステップS4でその擬似MDS行列を用いた暗号化アルゴリズム、すなわちFeistel構造が生成され、ステップS5でその暗号化アルゴリズムを用いて平文が暗号化されて、処理を終了する。

【0042】ステップS2で $A_T$ の値が0、または負の数となり、適切な線形変換が存在しないと判定されると、ステップS6でエラーが発生したことを示すメッセージが出力されて、処理を終了する。

【0043】図5は図4のステップS1、すなわちアクティブSボックス数の最大値 $A_T$ の計算処理の詳細フローチャートである。同図において処理が開始されると、まずステップS10で集合Tの内容が入力され、ステップS11で集合Tを構成するr個の要素のうちからk個を選んで生成された集合の要素の和の最小値 $u_k$ が $k=0, 1, 2, \dots, r$ に対して求められる。

【0044】続いてステップS12で、同様に集合Tの要素から任意のk個を選んで生成された集合の要素の和の最大値 $v_k$ が求められる。続いてステップS13で $k$  ( $=1, 2, \dots, r$ )と $k'$  ( $=0, 1, 2, \dots, r$ )に対して次の不等式

$$u_k \geq v_{k'} \quad (\text{ただし } v_0 = 0)$$

を満たす $k'$ の最大値を $k$ から減算した値が $w_k$  ( $k=1, 2, \dots, r$ )として求められる。

【0045】最後にステップS14で $w_k$ の最大値が $r+1$ から減算され、 $A_T$ の値として求められ、処理を終了する。図6は図4のステップS3の処理、すなわち擬

似MDS行列生成処理の詳細フローチャートである。同図において処理が開始されると、まずステップS20で分割ビット数の集合Tの内容に応じて、 $t_i$ 行、 $t_j$ 列であり、要素が0、または1となっている行列 $M_{ij}$  ( $i, j=1 \sim r$ )を作り、そのような $r \times r$ 個の行列 $M_{ij}$ を要素としてr行、r列に並べた行列Mがランダムに新しく選択される。図3で説明したF関数の例ではこの行列Mは全体としては32行、32列の行列となる。ここで $M_{ij}$ を行列Mの部分行列と呼ぶことにする。

【0046】続いてステップS21でeの値が1に初期化され、ステップS22でeの値がアクティブSボックス数の最大値 $A_T$ から1を引いた値を越えたか否かが判定され、越えていない場合にはステップS23で次式によって $c(e)$ の値が求められる。

$$【0047】 c(e) = e + r - A_T + 1$$

ステップS24で集合Tからe個の要素を任意に選び、集合 $T_1$ が新しく求められ、ステップS25で新しい集合 $T_1$ が選択できたか否かが判定され、選択できた場合にはステップS26で同様に集合Tから $c(e)$ 個の要素が任意に新しく選ばれ、集合 $T_2$ が求められ、ステップS27でそのような新しい集合 $T_2$ が選択できたか否かが判定される。なおステップS24、およびS26で新しく選択された集合 $T_1$ 、および $T_2$ を以下のように記述するものとする。

$$【0048】 T_1 = \{t_{i1}, t_{i2}, \dots, t_{ie}\}$$

$$T_2 = \{t_{j1}, t_{j2}, \dots, t_{jc(e)}\}$$

ステップS27で新しく集合 $T_2$ が選択できたと判定されると、ステップS28で行列Mの小行列のうちで集合 $T_1, T_2$ に対応する小行列のランク(階数)が求められる。この $T_1, T_2$ に対応する小行列の意味については後述する。そしてステップS29で求められたランクの値が 外1 または 外2 のどちら

【0049】

【外1】

$$\sum_{p=1}^e t_{ip}$$

【0050】

【外2】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【0051】か、すなわち行数と列数のいずれかに等しいか否かが判定される。等しい場合には、ステップS30で行列Mの小行列のうちで集合 $T_2, T_1$ に対応する小行列のランクが求められ、ステップS31でそのランクの値が 外3

【0052】

【外3】

$$\sum_{p=1}^e t_{ip}$$

【0053】または 外4 のいずれかに等しいか否かが判定される。

【0054】

【外4】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【0055】ステップS31でランクの値が2つの総和(行数、列数)のいずれかに等しいと判定されると、ステップS26に戻りc(e)個の要素が新たに選択され、新しい集合T<sub>2</sub>が求められ、ステップS27の判定以降の処理が繰り返される。

【0056】そしてステップS27でc(e)個の集合T<sub>2</sub>が新しく選択できなかったと判定されると、以前にステップS24で選択された集合、すなわちe個の要素からなる集合T<sub>1</sub>に対応する処理が終了したことになるため、ステップS24でe個の要素からなる集合T<sub>1</sub>として新しい集合が求められ、ステップS25以降の処理が繰り返される。

【0057】ステップS25で新しい集合T<sub>1</sub>が選択できなかったと判定されると、ステップS21で初期化されたe=1の値に対応する処理が終了したことになるので、ステップS32でeの値がインクリメントされ、ステップS22以降の処理が繰り返される。

【0058】このような処理の間に、ステップS29でランクの値が2つの総和の値のいずれにも等しくないと判定された時、およびステップS31で同様にランクの値が2つの総和のいずれにも等しくないと判定された時には、ステップS20でランダムに選択された行列Mが擬似MDS行列としては不適当なものであるものとして、ステップS20で新しい行列Mをランダムに選択する処理以降の処理が繰り返され、ステップS22でeの値がA<sub>T</sub>-1の値を越えたと判定されると、行列Mの内容が擬似MDS行列として出力され、処理を終了する。

【0059】図5、および図6で説明した処理について、具体例を用いて更に説明する。図3で説明した全部で32ビットの入力ビットに対する6個のSボックスに対応して、分割される入出力ビット数の集合は次式で与えられる。

【0060】T={6, 5, 5, 5, 5, 6}  
このような集合Tに対応して前述の最小値u<sub>k</sub>、および最大値v<sub>k</sub> (v<sub>k</sub>')は次のようになる。

【0061】(u<sub>1</sub>, u<sub>2</sub>, u<sub>3</sub>, u<sub>4</sub>, u<sub>5</sub>, u<sub>6</sub>)  
=(5, 10, 15, 20, 26, 32)

(v<sub>0</sub>, v<sub>1</sub>, v<sub>2</sub>, v<sub>3</sub>, v<sub>4</sub>, v<sub>5</sub>, v<sub>6</sub>)=  
(0, 6, 12, 17, 22, 27, 32)

その結果w<sub>k</sub>は次式となり、その最大値は1となる。

【0062】(w<sub>1</sub>, w<sub>2</sub>, w<sub>3</sub>, w<sub>4</sub>, w<sub>5</sub>, w<sub>6</sub>)  
=(1, 1, 1, 1, 1, 0)

最終的にアクティブSボックス数の最大値A<sub>T</sub>は、このw<sub>k</sub>の最大値を用いて次式によって求められる。

【0063】A<sub>T</sub>=(6+1)-1=6

このA<sub>T</sub>の値が6、すなわち正の数であることから、このように入出力ビット数が分割された6個のSボックスによる非線形変換に適切な線形変換が存在するということが判定される。前述のようにこの行列Mは全体として32行、32列であり、その要素が0、または1のうちからランダムに選択され、選択された行列が図6のフローチャートによって擬似MDS行列の性質を満たすかが判定される。

【0064】従ってそのような行列Mを生成するためには、原理的には32行、32列の行列の全ての要素を0、または1とした場合について図6のフローチャートの処理を繰り返して、擬似MDS行列を求めればよいことになるが、その計算量は膨大となる。

【0065】本実施形態では計算量を削減するための擬似MDS行列生成法を用いることにするが、その方法については後述することとし、その方法によって求められた行列Mの例を図7に示す。この例の行列が図6のフローチャートの処理において、最終的にステップS33で出力されるまでの過程の最初の部分について具体的に説明する。なお図7において、行列内の実線で区切られた部分は、図6のステップS20で説明した行列M内の部分行列M<sub>ij</sub>に相当する。

【0066】図6に対応する処理の具体例を説明する前に、まず例えばステップS28で説明したT<sub>1</sub>, T<sub>2</sub>に対応する小行列の意味について、図8を用いて説明する。図8において、例えば集合T<sub>1</sub>={t<sub>2</sub>, t<sub>3</sub>, t<sub>6</sub>}, T<sub>2</sub>={t<sub>2</sub>, t<sub>3</sub>, t<sub>5</sub>, t<sub>6</sub>}とした場合には、T<sub>1</sub>, T<sub>2</sub>に対応する小行列として図8(a)に示される行列が生成され、そのランクが求められる。すなわちそれぞれが行列であるM<sub>ij</sub>を部分行列とする行列Mから計3行と4列とが指定されて小行列が構成される。この小行列はビット単位、すなわち0、または1の要素単位では16行、21列の行列となる。

【0067】また図6のステップS30で説明したT<sub>2</sub>, T<sub>1</sub>に対応する小行列としては行として集合T<sub>2</sub>の要素であるt<sub>2</sub>, t<sub>3</sub>, t<sub>5</sub>、およびt<sub>6</sub>に相当する行と、集合T<sub>1</sub>の要素としてのt<sub>2</sub>, t<sub>3</sub>、およびt<sub>6</sub>に対応する列が選択されて小行列が構成される。この小行列を図8(b)に示す。この行列は21行、16列の行列である。

【0068】ここで本実施形態におけるMDS変換としての擬似MDS行列が持つべき性質について説明する。n=32ビットを6個に非均等に分割した集合の例としての前述のTに対して、アクティブSボックスの数の最大値はA<sub>T</sub>=6となる。これに対してビット数の分割を



均等に行う場合に $A_T$ に相当する値は7であり、その差は1となる。

【0069】前述のようにビット分割が均等な場合のMDS変換としてのMDS行列では、図8で説明したような $M_{ij}$  (全ての行数、および全ての列数は等しい) を要素とする行列から、任意の1行と1列を指定した(1, 1)小行列、2行と2列を指定した(2, 2)小行列、3行と3列を指定した(3, 3)小行列、...を考え、そのような任意の小行列が全て正則であることがMDS行列の性質として成立する。

【0070】これに対して擬似MDS行列では、前述の差が1であることから、ビット分割が均等な場合に選択される小行列の行、または列のいずれかに1を加えた行列が小行列として選択され、任意の小行列のランクがフル、すなわち小行列のランクがその行数、または列数に等しくなるという性質がある。

【0071】すなわち任意の(1, 2), (2, 1), (2, 3), (3, 2), (3, 4), (4, 3), (4, 5), (5, 4), (5, 6)、および(6, 5)の10種類の小行列のランクが、それぞれの小行列の行数、または列数に等しい行列を、擬似MDS行列として図6のフローチャートにおいて選択すべきことになる。これが本実施形態における擬似MDS行列が持つべき性質であるが、その詳細な数学的説明(証明など)については省略する。

【0072】ここで前述の例に戻り、図6のフローチャートに対応して、そのような性質を持つ行列Mの選択の過程の最初の部分の説明を続ける。まず図6のステップS21でeの値が1とされ、ステップS23で $c(e)$ の値として2が求められる。そしてステップS24で集合 $T_1$ として1個だけの要素を持つ $\{t_1\} = \{6\}$ が選択されたとする。またステップS26で $c(e)$ 、すなわち2個の要素を持つ集合 $T_2$ として $\{t_1, t_2\} = \{6, 5\}$ が選択されたものとする。

【0073】図9はこの場合にそのランクが計算されるべき、ステップS28における $T_1, T_2$ に対応する行列である。すなわち図8において行としては1行目、列としては1列目と2列目とが指定されることになり、小行列は $M_{11}$ と $M_{12}$ を成分とする行列であり、その実際の内容は図7から図9のようになる。この小行列のランクは6である。

【0074】このランクの値、すなわち6はステップS29で 外5 または 外6 のい

【0075】

【外5】

$$\sum_{p=1}^e t_{ip}$$

【0076】

【外6】

$$\sum_{q=1}^{c(e)} t_{jq}$$

【0077】いずれかの値と等しいか否かが判定される。これらの2つの値は図9の小行列の行数と列数を示し、この場合は行数、すなわち 外7 がランクの値と等しくなり

【0078】

【外7】

$$\sum_{p=1}^e t_{ip}$$

【0079】、この小行列はフルランクであることが判定される。図10はステップS30でそのランクが計算されるべき $T_2, T_1$ に対応する小行列の例である。前述と同様に、ここでは図8の $M_{ij}$ のうち、行としては1行目と2行目、例としては1列目が指定されることにより、 $M_{11}$ と $M_{21}$ とによって図10に示す小行列が構成される。そのランクは6であり、ステップS31でステップS29におけると同様に2つの総和と比較され、 外8 の値と等しいこ

【0080】

【外8】

$$\sum_{p=1}^e t_{ip}$$

【0081】とが判定されて、以後の処理が続けられる。そして図6のフローチャートに従って、前述の10種類の小行列の任意のものについて、各小行列のランクがフルであることが図7の32行、32列の行列に対して確認され、最終的にステップS33でこの行列Mが擬似MDS行列として出力されることになる。

【0082】次に図7に示した擬似MDS行列の生成法について説明する。この行列を生成するためには、原理的には前述のように32行×32列の行列の全ての要素を0、または1にランダムに変化させて、図6のフローチャートを満足する行列Mを探すことになるが、その計算量は膨大となる。

【0083】そこでより能率的な方法として、本実施形態においてはまず全ビット数を30ビットとし、30ビットを6個に分割した集合 $T = \{5, 5, 5, 5, 5, 5\}$ に対するMDS行列を従来技術によって求め、求められた30行、30列の行列に対して図7に示すように最も上の行の $M_{1j}$  ( $j=1\sim6$ )、最も下の行の $M_{6j}$  ( $j=1\sim6$ )、最も左の列の $M_{i1}$  ( $i=1\sim6$ )、および最も右の列の $M_{i6}$  ( $i=1\sim6$ )に対応してそれぞれ1行、1列の要素を追加することで、擬似MDS行列を作成することにする。

【0084】図11、図12はそのような30行、30列のMDS行列を構成するための5行、5列の部分行列



3 2 個を示している。この3 2 個の部分行列はそれぞれ5 行、5 列の行列であり、各行列には0 ~ 3 1 の番号が付けられている。0 番の行列は図1 1 の左上の行列であり、5 行、5 列の行列の要素は全て0 である。5 行、5 列の下の“0”はこの行列に対応する(同じ配置の)行列式の値を示している。0 番目の行列に対しては、当然対応する行列式の値は0 である。

【0085】例えばその下の番号1 の行列に対する行列式の値は1 であり、以降図1 2 の右下の3 1 番までの全ての行列に対する行列式の値も1 となっている。従来技術の方法を用いることによって図1 1、図1 2 に示した番号の5 行、5 列の部分行列を6 行、6 列に並べることによって、3 0 ビットを6 個に均等分割した場合に対応するMDS 行列の例として図1 3 の行列が得られる。行列内の数字は図1 1、図1 2 で説明した各行列の番号を表わす。

【0086】図1 3 に示した行列は3 0 行、3 0 列の行列であり、最も上、下の部分行列に対して1 行、最も左、右の部分行列に対して1 列の要素をランダムに追加し、その行列に対して図6 で示したフローチャートの処理を実行することによって、図7 に示した擬似MDS 行列を比較的容易に生成することができる。

【0087】図1 4 は本発明を実現するためのプログラムのコンピュータへのローディングの説明図である。本発明の実施形態としての暗号化装置、例えば図2 に示したシステムなどは、当然一般的なコンピュータシステムとして構成することができる。

【0088】図1 4 はそのようなシステムの構成を示し、コンピュータ3 1 は本体3 2 と、メモリ3 3 とによって構成されている。メモリ3 3 はランダムアクセスメモリ(RAM)、ハードディスク、磁気ディスクなどの記憶装置であり、本発明の特許請求の範囲第1 0 項のプログラムや、図4 ~ 図6 で説明したプログラムなどはメモリ3 3 に格納され、そのプログラムが本体3 2 によって実行されることにより、本発明の擬似MDS 行列が求められ、入力データに対する暗号化が行われる。

【0089】本発明を実現するためのプログラムは、プログラム提供者側からネットワーク3 4 を介してコンピュータ3 1 にロードされることも、また市販され、流通している可搬型記憶媒体3 5 に格納され、そのプログラムがコンピュータ3 1 にロードされることによって実現されることも可能である。可搬型記憶媒体3 5 としてはフロッピー(登録商標)ディスク、CD-ROM、光ディスク、光磁気ディスクなど、様々な形式の記憶媒体を使用することができる。前述のプログラムなどは、このような記憶媒体に格納され、コンピュータ3 1 にロードされることによって、本実施形態における擬似MDS 行列が生成され、その行列を用いて入力データに対する暗号文を生成することが可能となる。

【0090】

【発明の効果】以上詳細に説明したように、本発明によれば、F 関数の内部の複数のS ボックスの入出力サイズが同一でない場合において、適切な線形変換としての擬似SMD 行列の存在の有無を判定することができ、そのような行列が存在する場合にはその擬似MDS 行列を生成し、その行列を使用した暗号化を行うことによって、拡散性能に優れた暗号を生成することができ、暗号化装置の性能向上に寄与するところが大きい。

【図面の簡単な説明】

【図1】本発明の原理構成ブロック図である。

【図2】本発明の実施形態としての暗号化装置のシステム構成を示すブロック図である。

【図3】本実施形態におけるF 関数の構成例を示す図である。

【図4】暗号文生成の全体処理フローチャートである。

【図5】アクティブS ボックスの数の最大値 $A_T$ を求める処理の詳細フローチャートである。

【図6】擬似MDS 行列を求める処理の詳細フローチャートである。

【図7】求められた擬似MDS 行列の例を示す図である。

【図8】2 つの集合に対応する小行列を説明する図である。

【図9】擬似MDS 行列の小行列の例(その1)を示す図である。

【図10】擬似MDS 行列の小行列の例(その2)を示す図である。

【図11】3 0 行×3 0 列のMDS 行列を求めるための部分行列を示す図(その1)である。

【図12】3 0 行×3 0 列のMDS 行列を求めるための部分行列を示す図(その2)である。

【図13】図1 1、図1 2 の部分行列を用いたMDS 行列の例を示す図である。

【図14】本発明におけるプログラムのコンピュータへのローディングを説明する図である。

【図15】DES 暗号の基本構造を示す図である。

【図16】図1 5 におけるF 関数の構成例の説明図である。

【図17】F 関数内の線形変換P としてのMDS 変換の説明図である。

【図18】MDS 変換としてのMDS 行列の説明図である。

【符号の説明】

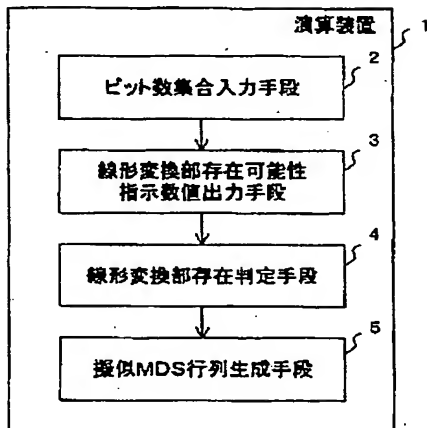
- 1 演算装置
- 2 ビット数集合入力手段
- 3 線形変換部存在可能性指示数値出力手段
- 4 線形変換部存在判定手段
- 5 擬似MDS 行列生成手段
- 10 処理装置
- 11 入力ファイル

- 12 出力ファイル  
13 表示装置  
14 入出力装置  
15  $A_T$  計算部

- 16 線形変換部存在判定部  
17 擬似MDS行列生成部  
18 暗号文生成部

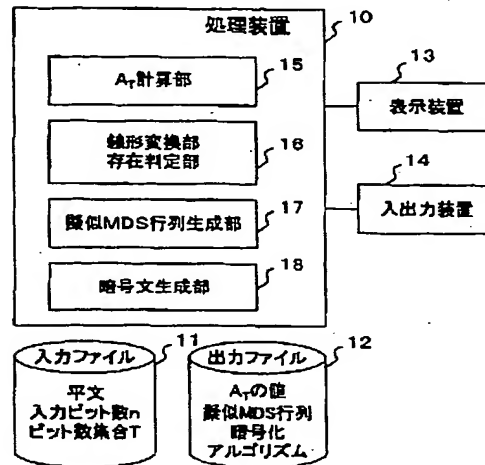
【図1】

本発明の原理構成ブロック図



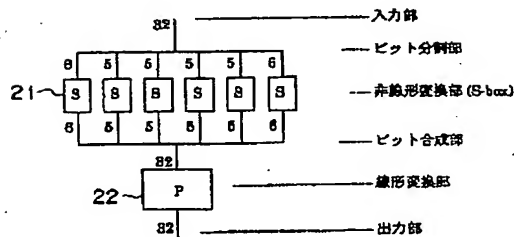
【図2】

本発明の実施形態としての暗号化装置のシステム構成を示すブロック図



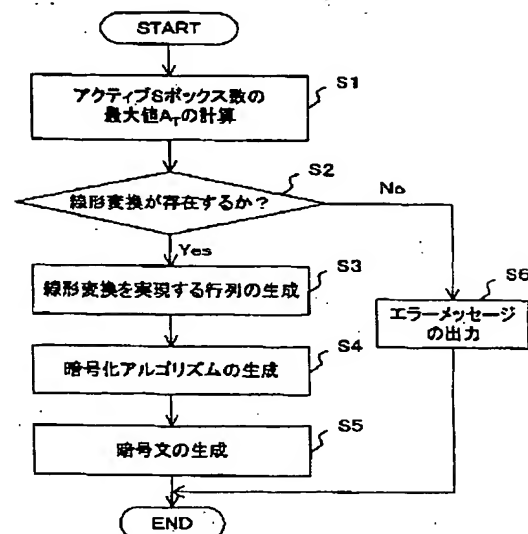
【図3】

本実施形態におけるF関数の構成例を示す図



【図4】

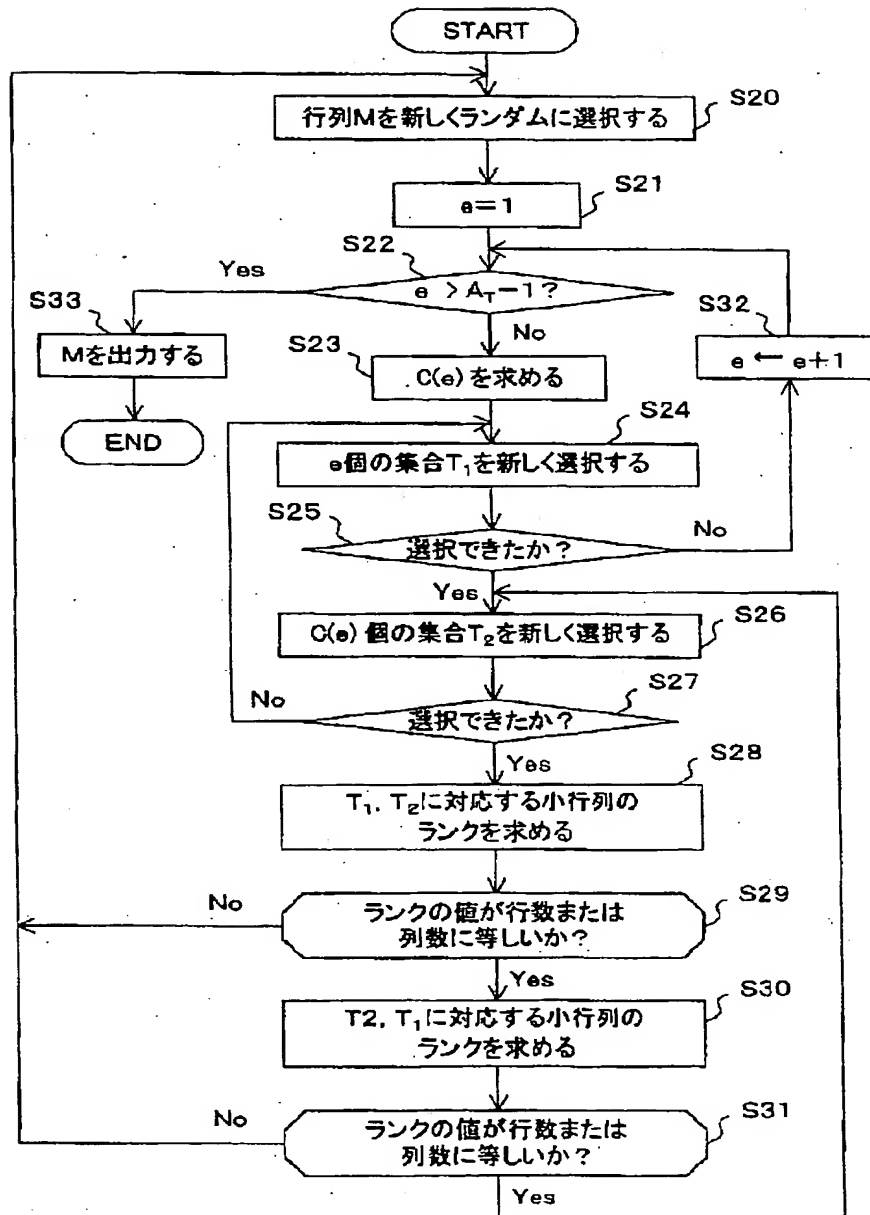
暗号文生成の全体処理フローチャート





【図6】

擬似MDS行列を求める処理の詳細フローチャート



【図8】

2つの集合に対応する小行列を説明する図

$$M = \begin{pmatrix} 6 & 5 & 5 & 5 & 5 & 6 \\ M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} & M_{36} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ M_{61} & M_{62} & M_{63} & M_{64} & M_{65} & M_{66} \end{pmatrix} \Rightarrow \begin{pmatrix} M_{22} & M_{23} & M_{25} & M_{26} \\ M_{32} & M_{33} & M_{35} & M_{36} \\ M_{62} & M_{63} & M_{65} & M_{66} \end{pmatrix}$$

(a)

$$M = \begin{pmatrix} 6 & 5 & 5 & 5 & 5 & 6 \\ M_{11} & M_{12} & M_{13} & M_{14} & M_{15} & M_{16} \\ M_{21} & M_{22} & M_{23} & M_{24} & M_{25} & M_{26} \\ M_{31} & M_{32} & M_{33} & M_{34} & M_{35} & M_{36} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} & M_{46} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} & M_{56} \\ M_{61} & M_{62} & M_{63} & M_{64} & M_{65} & M_{66} \end{pmatrix} \Rightarrow \begin{pmatrix} M_{22} & M_{23} & M_{26} \\ M_{32} & M_{33} & M_{36} \\ M_{52} & M_{53} & M_{56} \\ M_{62} & M_{63} & M_{66} \end{pmatrix}$$

(b)

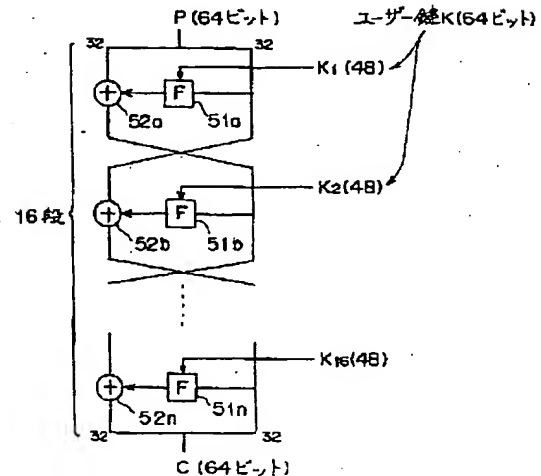
【図10】

擬似MDS行列の  
小行列の例(その2)を示す図

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

【図15】

DES 暗号の基本構造を示す図



【図11】

30行 x 30列のMDS行列を求めるための部分行列を示す図

(その1)

0 : matrix[5,5]=	8 : matrix[5,5]=
00000	01000
00000	10000
00000	00101
00000	01010
00000	10100
0	1
1 : matrix[5,5]=	9 : matrix[5,5]=
00001	01001
00010	10010
00100	00001
01000	00010
10000	00100
1	1
2 : matrix[5,5]=	10 : matrix[5,5]=
00010	01010
00100	10100
01000	01101
10000	11010
00101	10001
1	1
3 : matrix[5,5]=	11 : matrix[5,5]=
00011	01011
00110	10110
01100	01001
11000	10010
10101	00001
1	1
4 : matrix[5,5]=	12 : matrix[5,5]=
00100	01100
01000	11000
10000	10101
00101	01111
01010	11110
1	1
5 : matrix[5,5]=	13 : matrix[5,5]=
00101	01101
01010	11010
10100	10001
01101	00111
11010	01110
1	1
6 : matrix[5,5]=	14 : matrix[5,5]=
00110	01110
01100	11100
11000	11101
10101	11111
01111	11011
1	1
7 : matrix[5,5]=	15 : matrix[5,5]=
00111	01111
01110	11110
11100	11001
11101	10111
11111	01011
1	1

## 【図12】

30行 x 30列のMDS行列を求めるための部分行列を示す図

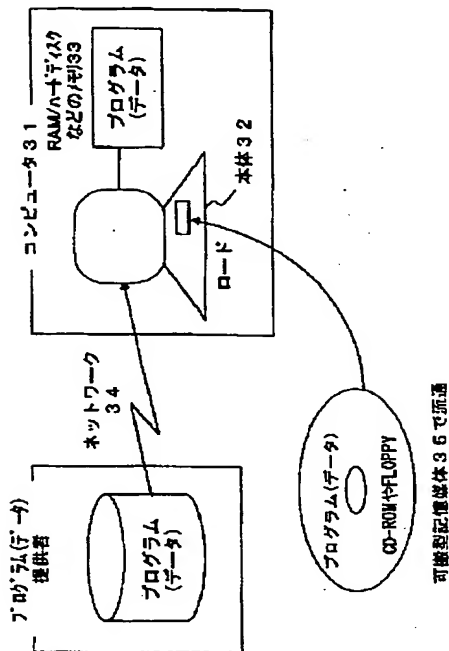
(その2)

16 : matrix[5,5]=	24 : matrix[5,5]=
10000	11000
00101	10101
01010	01111
10100	11110
01101	11001
1	1
17 : matrix[5,5]=	25 : matrix[5,5]=
10001	11001
00111	10111
01110	01011
11100	10110
11101	01001
1	1
18 : matrix[5,5]=	26 : matrix[5,5]=
10010	11010
00001	10001
00010	00111
00100	01110
01000	11100
1	1
19 : matrix[5,5]=	27 : matrix[5,5]=
10011	11011
00011	10011
00110	00011
01100	00110
11000	01100
1	1
20 : matrix[5,5]=	28 : matrix[5,5]=
10100	11100
01101	11101
11010	11111
10001	11011
00111	10011
1	1
21 : matrix[5,5]=	29 : matrix[5,5]=
10101	11101
01111	11111
11110	11011
11001	10011
10111	00011
1	1
22 : matrix[5,5]=	30 : matrix[5,5]=
10110	11110
01001	11001
10010	10111
00001	01011
00010	10110
1	1
23 : matrix[5,5]=	31 : matrix[5,5]=
10111	11111
01011	11011
10110	10011
01001	00011
10010	00110
1	1



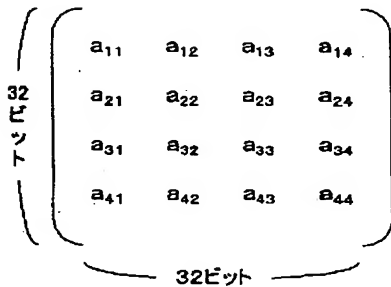
【図14】

本発明におけるプログラムの  
コンピュータへのローディングを説明する図



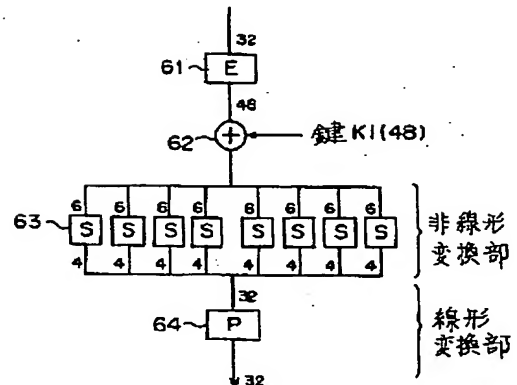
【図18】

MDS変換としてのMDS行列の説明図



【図16】

図15におけるF関数の構成例の説明図



フロントページの続き

(72)発明者 武仲 正彦  
神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内  
(72)発明者 鳥居 直哉  
神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(72)発明者 矢嶋 純  
神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内  
(72)発明者 屋並 仁史  
神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(72)発明者 横山 和弘  
神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

Fターム(参考) 5J104 AA19 JA13 JA17 NA02